NASA-TM-100442 19880015823

NASA Technical Memorandum 100442

Application of Flight Systems Methodologies to the Validation of Knowledge-Based Systems

Eugene L. Duke

July 1988

National Aeronautics and
Space Administration

NASA Technical Memorandum 100442

Application of Flight Systems Methodologies to the Validation of Knowledge-Based Systems

Eugene L. Duke
Ames Research Center, Dryden Flight Research Facility, Edwards, California

1988

National Aeronautics and
Space Administration
Ames Research Center
Dryden Flight Research Facility
Edwards, California 93523-5000

APPLICATION OF FLIGHT SYSTEMS METHODOLOGIES TO THE VALIDATION OF KNOWLEDGE-BASED SYSTEMS

Eugene L. Duke
NASA Ames Research Center
Dryden Flight Research Facility
Edwards, California

ABSTRACT

Flight and mission-critical systems are verified, qualified for flight, and validated using well-known and well-established techniques. These techniques define the validation methodology used for such systems. In order to verify, qualify, and validate knowledge-based systems (KBSs), the methodology used for conventional systems must be addressed, and the applicability and limitations of that methodology to KBSs must be identified. The author presents an outline of how this approach to the validation of KBSs is being developed and used at the Dryden Flight Research Facility of the NASA Ames Research Center.

1 INTRODUCTION

The verification and validation (V&V) of flight-critical systems is a major activity at the Dryden Flight Research Facility of the NASA Ames Research Center (Ames-Dryden). The Ames-Dryden staff assumes safety-of-flight responsibilities for all vehicles flown at the facility. Because these systems are used in research aircraft, the V&V experience at Ames-Dryden is primarily with one-of-a-kind research systems on experimental vehicles. While the range of this experience at Ames-Dryden is somewhat more narrow than that of the validation of flight-critical systems for commercial operations [1], this experience is directly applicable to the types of knowledge-based systems (KBSs) within NASA research programs, whose requirements are to qualify and validate unique, one-of-a-kind research systems.

The Ames-Dryden V&V methodology for embedded flight-critical systems relies on testing, peer review, abstract models, simulations, and flight validation. This methodology also relies, in large part, on engineering judgment and a tradition that has evolved from the experience with flight-critical systems from the first simplex digital aircraft flight control system on the F-8 digital fly-by-wire (DFBW) aircraft [2], through the triplex DFBW system on the F-8 aircraft [3,4], the 3/8th scale F-15 remotely piloted research vehicle (RPRV) [5], the highly maneuverable aircraft technology (HiMAT) vehicle [6,7,8,9], and the advanced fighter technology integration program AFTI/F-16 [10,11], to the X-29 forward-swept wing aircraft. The result of this evolving, hands-on development of qualification and V&V methodologies is a practical approach that maximizes safety and allows system qualification, verification, and validation to proceed in an expeditious and resource-effective manner.

The V&V methodology used at Ames-Dryden is the same methodology that has actually been used for all flight-critical control systems in non-commercial aeronautical flight vehicles, including the F-18, Space Shuttle, and B-1 aircraft. This methodology uses a subset of the V&V techniques in use or advocated within the aeronautics community. The larger issues of certification and the validation of highly reliable, fault-tolerant systems have been of lesser concern than those of qualifying and conducting flight validation of flight-critical systems.

The basic methodology for the V&V of conventional operation-critical systems is directly applicable to the V&V of KBSs. In fact, if KBSs are to be used in operation-critical applications, the qualification of these KBSs will have to be

performed within the context of established procedures and will have to address the requirements placed upon the qualification of conventional operation-critical systems. Thus, it is essential that the main features of this well-established V&V methodology be understood.

NOMENCLATURE

a_n	normal acceleration, g
h	altitude, ft
$\dot{h}$,	altitude rate, ft/sec
$\ddot{h}$	altitude acceleration, ft/sec^2
$f(\ddot{h})$	functional relationship between $\ddot{h}$ and a_n
K_a	aerodynamic gain
K_D	proportional path gain
K_I	integral path gain
δ_{e_x}	equivalent longitudinal stick command
Δh	difference between desired and actual altitude
$\frac{1}{s}$	representation of integrator in Laplace variable notation

Abbreviations and Acronyms

AFSR	airworthiness and flight safety review
AFTI	advanced fighter technology integration
AI	artificial intelligence
Ames-Dryden	Dryden Flight Research Facility of the NASA Ames Research Center
CCB	configuration control board
CCR	configuration change request
CDR	critical design review
DFBW	digital fly-by-wire
DR	discrepancy report
FRR	flight readiness review
HiMAT	highly maneuverable aircraft technology
KBS	knowledge-based system
PC	program change (software)
PDR	preliminary design review
RPRV	remotely piloted research vehicle
QA	quality assurance
SDR	system design review
STR	system test report
V&V	verification and validation
WO	work order (hardware)

Definitions

The majority of the following definitions is taken verbatim from Szalai and others [4].

certification The determination by a regulatory authority that a product meets the regulations for that product.

embedded system A system that is an integral part of some larger system. This distinction is particularly important when a subsystem interacts with the larger system in such a way that a failure in the embedded system can propagate to the larger system or cause the larger system to fail.

fault tolerant A system which is able to continue to provide critical functions after the occurrence of a fault.

flight critical A component or system whose failure could cause loss of the aircraft.

mission critical A component or system whose failure could result in the inability to perform a mission.

operation critical A component or system whose failure could result in loss of the aircraft, loss of life or limb, compromise public safety, result in substantial financial loss, or inability to perform a mission.

qualification A formal process whereby a system or aircraft is defined to be ready for flight operations.

system An entity of fixed identity united by some form of purpose, interaction, or interdependence that can be meaningfully isolated.

validation The determination that a resulting product meets the objectives that led to the specification for the product. This determination usually includes operation in a real environment.

verification The determination that a design meets the specification. Verification is usually a part of the validation process. A simulated environment is often used.

2 A METHODOLOGY FOR CONVENTIONAL, EMBEDDED FLIGHT-CRITICAL CONTROL SYSTEMS

The basis of the Ames-Dryden flight qualification and V&V methodology for embedded flight-critical systems is the incremental verification of system components, integration testing, configuration management, and flight validation. The application of the verification, integration testing, and flight validation is discussed within the overall context of the system life cycle. The configuration management aspect of the qualification and V&V methodology is discussed separately.

2.1 Verification, Validation, and the System Life Cycle

Verification and validation is an ongoing process that is an integral part of the system life cycle. The system life cycle for conventional flight control systems is often characterized as a series of stages (figure 1).

When functional specifications are derived from the system goals and requirements, those specifications must be critically examined to establish that the specifications adequately address the system goals and requirements. Similarly, the design specifications must meet the functional specifications. This critical examination is accomplished by system design reviews (SDRs), preliminary design reviews (PDRs), and critical design reviews (CDRs). The SDR is a presentation and review of the conceptual design of the system; the goals and requirements are addressed and top-level definition of functional specifications are provided. The purpose of the SDR is to ensure that the understanding of the goals and requirements for the system is consistent between the requesters and designers. The PDR is a presentation of a first-order definition of the system design including a presentation of how the functional specifications are being addressed in the design. The CDR is a detailed design review in which a functional design is presented for review. (Theoretically, no hardware or software is to be implemented until after the CDR, but in practice, system components are implemented early in the life cycle, often before the SDR, to test ideas and may be directly incorporated or modified for the system as finally implemented.) At each of these reviews, designs are presented to a large audience with broad interests ensuring that system-level goals and requirements are addressed, user requirements are satisfied, and safety issues are adequately considered. The review boards provide detailed feedback to the system designers and implementers, and weaknesses or criticisms raised at a design review must be addressed at the next level of review.

The design review process is an iterative, and, one hopes, convergent process in which the goals

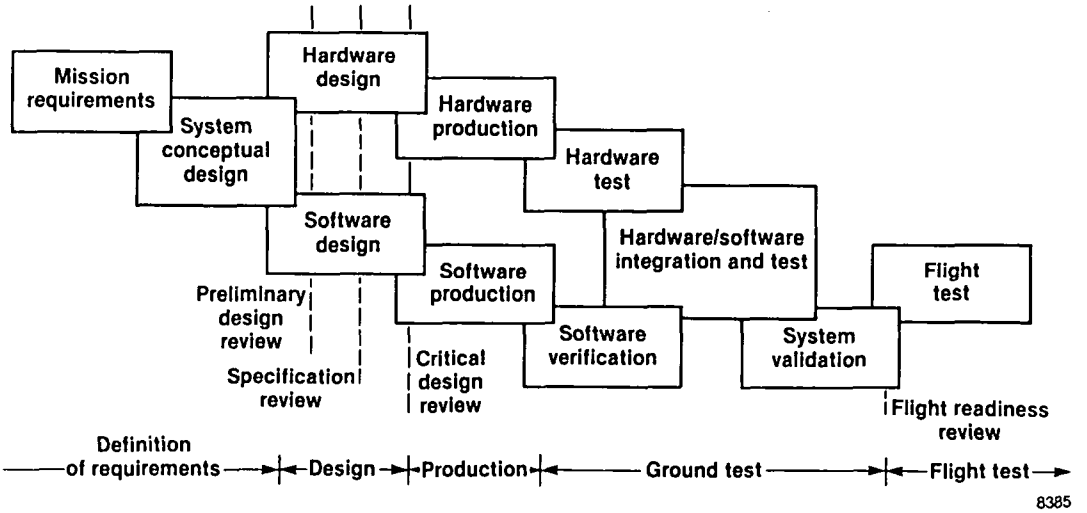


Figure 1. Ames-Dryden Life Cycle for Research Systems

and requirements and functional specifications are interpreted and translated into a design specification that, among other things, establishes a partitioning of functions between hardware and software. Design specifications for both hardware and software are translations of the functional specifications which in turn embody the system-level goals and requirements. The design specification is supported by an interface definition establishing both the interfaces between the system and its environment and the interfaces between the hardware and software.

The design specifications are transformed into hardware and software realizations. This transformation is not a straightforward, one-step process. The transformation of a design specification to an implemented prototype system requires the development and testing of numerous software procedures and hardware circuits, each of which is a prototype of some element in the larger system.

The implementation of system elements or components is supported by a variety of analysis tools and testing techniques [1, 2, 4, 12, 13]. The analysis tools used include failure modes and effects analysis, independent review, static verification, independent calculations, conjectures, and suspicions. This analysis is conducted on abstract models of the system or of the system components. Linear system models, aggregate system models, block diagrams, flow diagrams, schematics, source programs, specifications, and simulations are some of the main abstract models used. This analysis of abstract models is used to translate requirement and design specifications into a physical realization.

The physical realization of a system is constructed from physical realizations of system components such as circuits, microprocessors, computers, and software modules. Hardware components are often breadboard, brassboard, or nonflight-qualified versions of the actual flight system; these hardware components are bench tested in isolation and then incorporated into "hot-bench" test facilities that incorporate other simulated or actual physical systems. Typical of these hot-bench test facilities are simulations incorporating flight hardware (hardware-in-the-loop simulations) [6, 7, 8] or iron bird facilities based on extensive replication of flight systems and are often based on the use of decommissioned aircraft [4].

Simulation testing provides a closed-loop facility wherein the system is exposed to an environment that closely resembles the electronic and data environment in which the system must actually operate. Simulation also provides a facility for testing that the hardware and software of the system are integrated and operating together. The realism of the simulation is determined by the operating requirements for the flight application of the system (see section 2.3). Simulation is where the pilot (the system user) is first exposed to and allowed to evaluate the system.

This analysis, testing, and verification along with the configuration management process (see section 2.2) constitute main components of the qualification process wherein a system is determined to be ready for flight test. These results are presented to a flight readiness review (FRR) team composed of nonproject engineers from multiple engineering disciplines who perform an independent and in-depth review of the system design, analysis, test results, and configuration management. The FRR team is empowered to recommend additional analysis, testing, or documentation. The results of the FRR are presented at an airworthiness and flight safety review (AFSR) panel where the project team seeking authorization for approval to begin flight testing responds to the findings of the FRR. The AFSR panel is typically composed of engineering and operations managers and flight safety personnel. Only after the AFSR is satisfied is a system taken to flight.

Flight validation is an extension of the testing methods performed on physical models in the simulation. For a research system, the flight tested component is a physical model of itself; if the system is to be fielded in an operational environment, the flight tested system is often a prototype of the final system. During flight test, the system is exposed to the total physical, electronic, and data environment in which it is designed to operate.

Gault and others [1], Holt and others [12], and Hopkins [13] propose the use of additional abstract models (aggregate models) and analysis methods (formal proofs and statistical analysis). These models and analysis tools address one of the chief

limitations in the Ames-Dryden methodology: the reliance on testing, both failure modes and effects and nominal condition testing. This becomes a serious concern when considering either highly reliable, fault-tolerant systems or highly complex systems. Hartmann and others [13] describe the number of tests required for such systems as a *fundamental* problem:

> The fundamental problem of fault tolerance validation is the vast number of test cases when all possible combinations of flight conditions and multiple faults are considered.

This view is confirmed by Gerhart and others [1], Holt and others [12], and Gerhart [15], who claim that exhaustive testing is not possible for any but the simplest of systems.

2.2 Configuration Management

Configuration management is the orderly and systematic process of ensuring consistency in development, documentation, testing, problem reporting, and maintenance of a system. The use of a configuration control board (CCB) with review and change approval authority, consisting of representatives from several engineering disciplines, is a key feature of configuration management. Petersen and Flores [16] describe the configuration management process:

> The primary purpose of the software control and system configuration management process for flight-critical digital flight control systems is to provide a method for efficient flight system development and a procedure for assuring safe flight operations. The process is designed to control system configuration changes by managing the primary system development phases ... and to resolve discrepancies uncovered during system testing. In addition, the configuration control process prescribes stringent test and documentation requirements and provides for visibility of changes across all involved engineering disciplines through formal review procedures.

Petersen and Flores [16] also present block diagrams showing the steps in the software control and system configuration process (figure 2) and the documentation flow and tracking process (figure 3) used at Ames-Dryden. This process is initiated when the system is put under configuration control which is generally well into the system development cycle.

Figure 2 shows how new system requirements or anomalous system behavior are accommodated in the software control and system configuration management process. New system requirements are introduced into the configuration management process using a configuration change request (CCR); anomalous system behavior is recorded on a discrepancy report (DR) (figure 3).

The use of DRs to record any anomalous behavior is useful for identifying and correcting problems that result from operator error, initialization, or system design. Additionally, the extensive use of DRs provides a means of isolating incipient problems by identifying areas or functions in the system that are repeatedly involved in or associated with anomalous behavior. Tracking DRs also facilitates one aspect of the process of building confidence in the system (see section 2.1): it provides a means of judging the maturity of a system through experience with that system over an extended period. Typical experience with systems as a function of time is shown in figures 4 and 5.

It is important to note that the problems identified in every DR cannot be or are not always remedied. A problem that occurs in a test facility might be highly unlikely in flight; the subject of a DR might even be based on a rethinking of the system design that uncovers a failure mode or potential problem. Program schedule slippage or the costs of fixing the problem are often overriding concerns. The effect of problems identified in DRs is evaluated in terms of the risk associated with them. Those known problems that are identified on DRs and not remedied are called "accepted risks." Accepted risks are always clearly identified before flight testing. The risks associated with these problems are made visible to, and are evaluated by, independent reviewers such as those comprising the AFSR panel (see section 2.1).

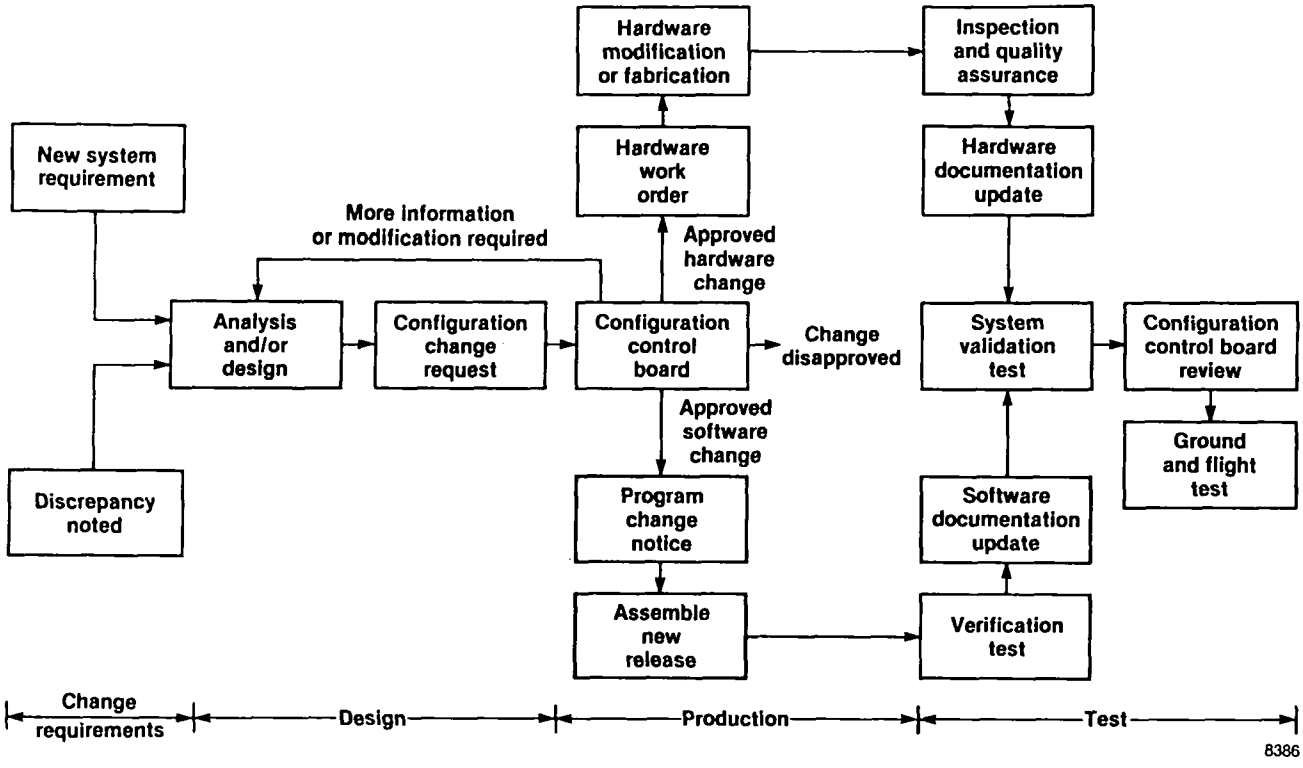


Figure 2. Software Control and System Configuration Management Process

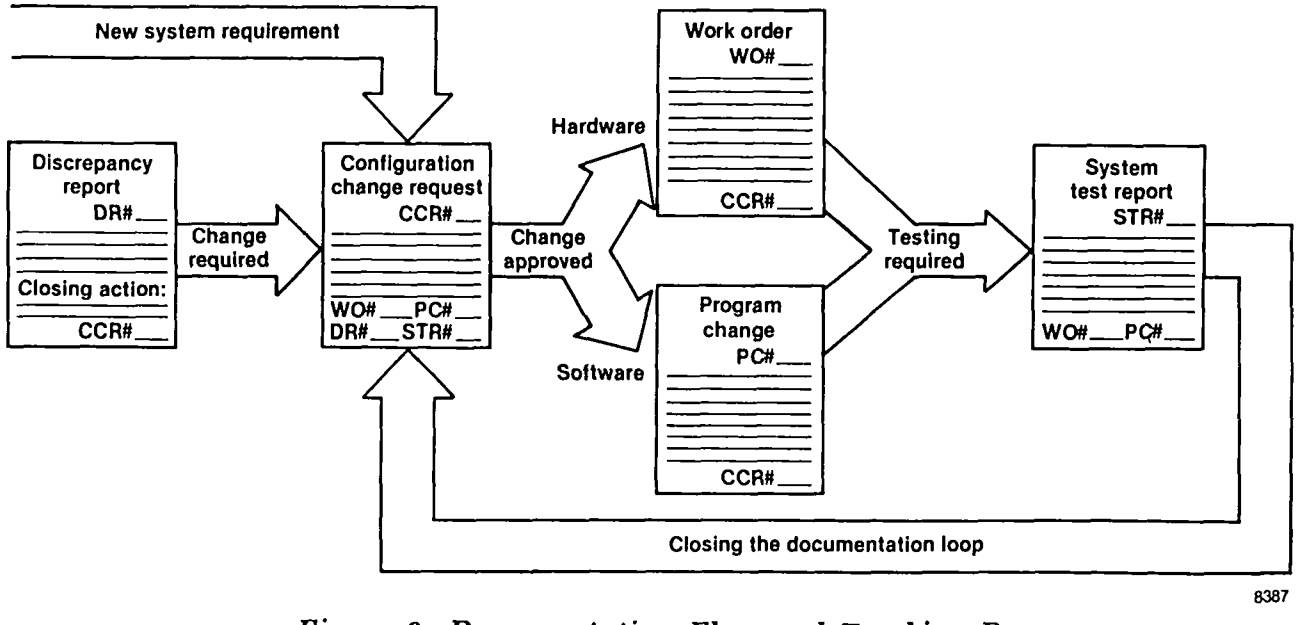


Figure 3. Documentation Flow and Tracking Process

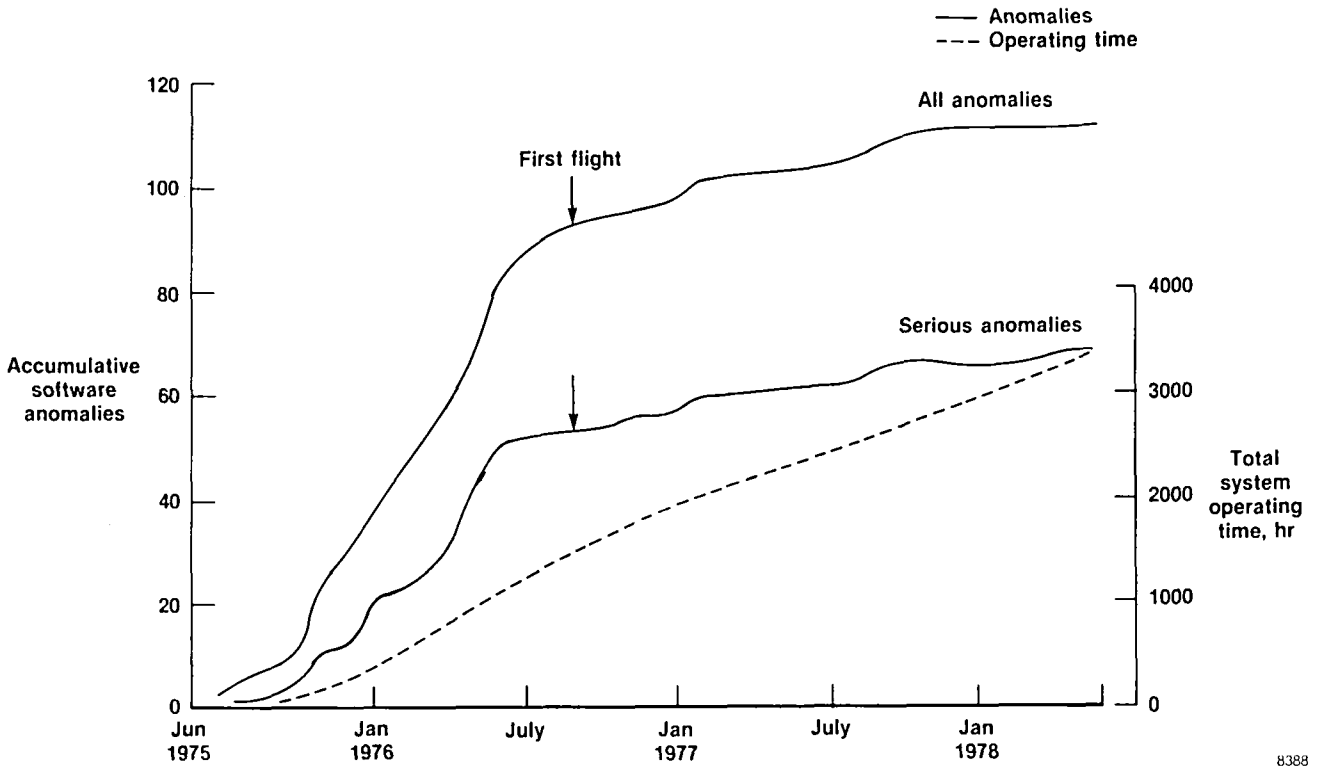


Figure 4. F-8 DFBW Software-Anomaly Experience

2.3 System Criticality and Its Impact on Validation

The requirements imposed on the V&V and configuration management process are determined by the criticality of the system. For aircraft flight systems, three levels of criticality are generally recognized:

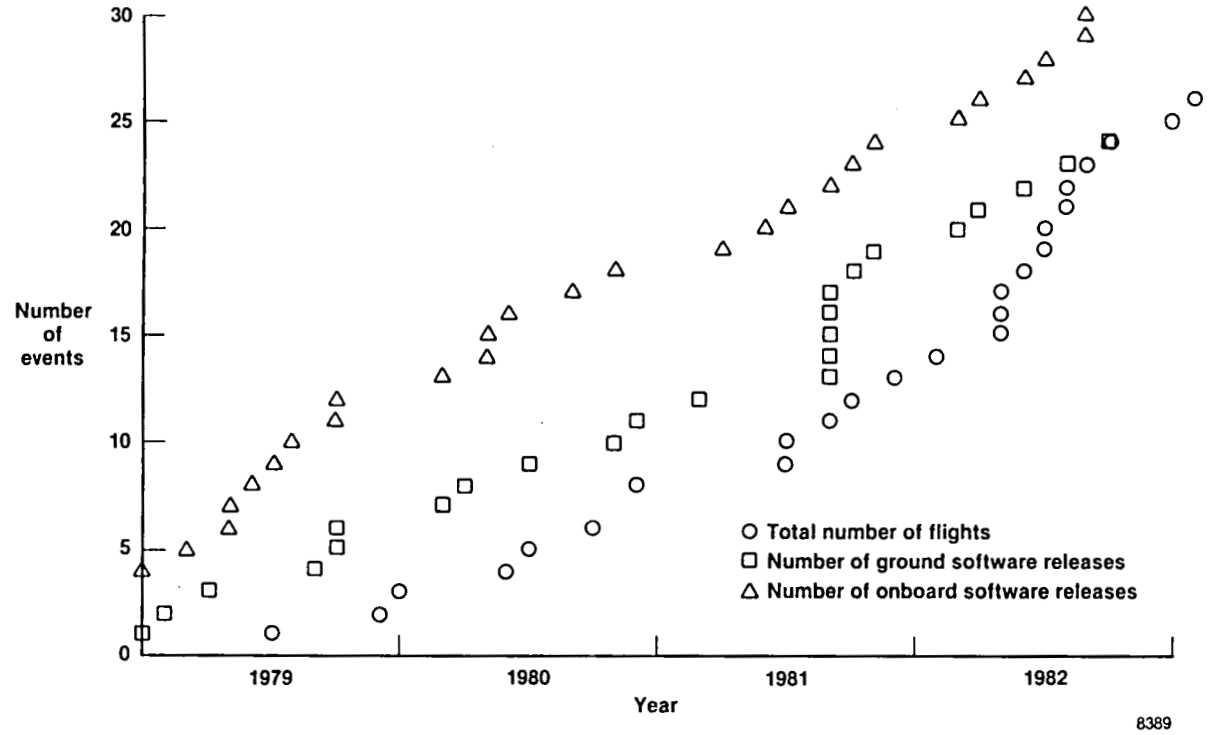
A. Systems whose failure could cause loss of life or limb, compromise public safety, or result in substantial financial loss;

B. Systems whose failure could cause mission failure (mission-critical);

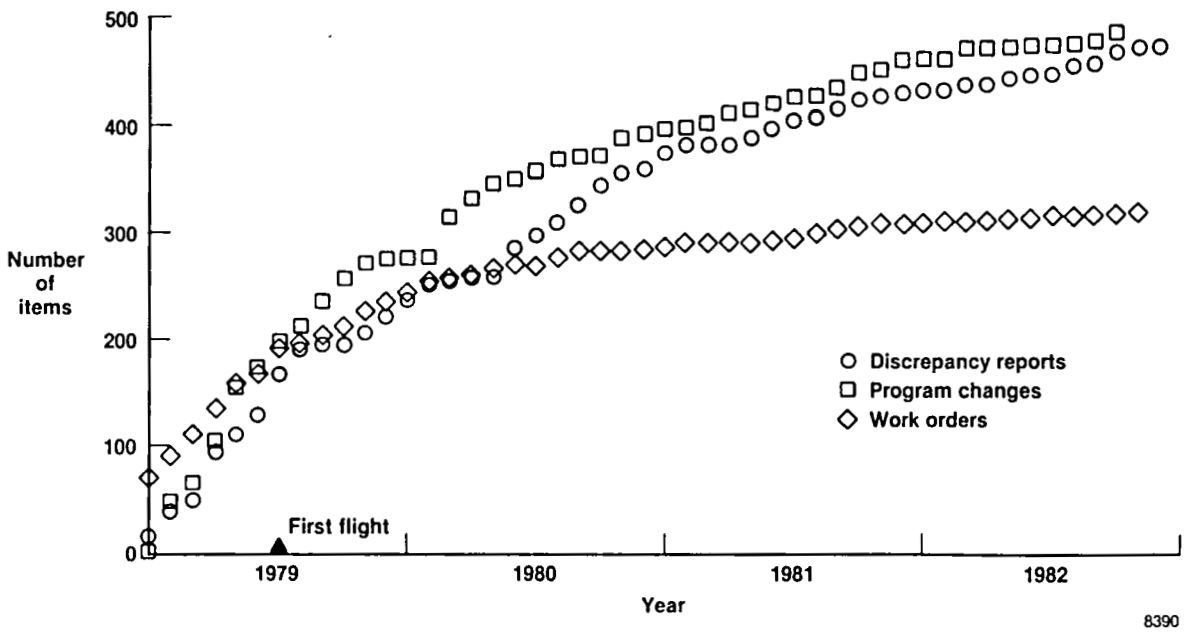
C. Systems whose failure could cause inaccurate results or inefficient use of resources.

The level of criticality of a system is primarily determined by those requesting that the system be developed. However, system designers or any of the independent review teams can modify that determination. In practice, we have found that it is usually easier and less work to classify a system as a level B rather than try to support and defend a classification at level C. Nontechnical factors are often taken into account when determining at which level to class a system. A system might be treated at level A when that system or the project of which it is a part is highly visible and any perceived problem might jeopardize research goals.

The configuration management process increases in formality at each higher level of criticality: the composition of the CCB is broader and consists of more members, the requirements for documentation increase, and testing requirements for system changes are more extensive. The requirements for the simulation also increase with higher levels of criticality: a level C system might be qualified off-line, using interactive or batch simulations and stand-alone hardware tests; a level B system requires at least a real-time, piloted simulation for closed-loop testing; and a level A system generally requires a hardware-in-the-loop simulation or an iron bird.



(a) Flights and software releases.



(b) Discrepancy reports, program changes, and work orders.

Figure 5. HiMAT System Development History

3 APPLICATION OF VALIDATION METHODOLOGY TO KNOWLEDGE-BASED SYSTEMS

The V&V methodology used for conventional, embedded operation-critical flight systems provides an established and accepted set of procedures upon which a methodology for KBSs can be based. While this position may be controversial in the AI community, the political and sociological realities of flight research and testing will ultimately dictate that any methodology for the validation of KBSs at least address the currently used methodology for conventional systems.

3.1 The Life Cycle Model for Knowledge-Based Systems

The proposed approach to the V&V of KBSs relies on the life cycle model shown in figure 1. The life cycle model for a KBS has been a topic of considerable concern to some who have addressed the validation of a KBS, and several models have been proposed [17, 18, 19]. These models stress the development and prototyping process in a KBS. The motivation for developing these models is apparently to address the lack of a clear or well-defined statement of system goals and requirements and to highlight the prototyping process common in the development of KBSs. While the proponents of these models would probably contend that there is a fundamental difference between the life cycle of a KBS and a conventional system, another view is that this apparent difference is more reflective of the maturity of KBSs rather than of anything fundamental.

Because KBSs are just emerging in operation-critical applications, there is little certainty of capabilities and limitations of these systems. The prototyping that is a common feature in the development of a KBS often represents an attempt to establish requirements for a given application. This definition of requirements, capabilities, and limitations through prototyping is not unlike that used in conventional systems when new techniques or applications are attempted. The difference is in the body of knowledge and experience behind the use of conventional systems as opposed to that for KBSs. Also reflected in this prototyping is the lack of maturity of artificial intelligence (AI) techniques in general that provides little basis for the selection of control and knowledge representation methods.

3.2 Problems in the Verification and Validation of Knowledge-Based Systems

There are several issues that are almost certain to create problems for anyone attempting to validate operation-critical KBSs. Perhaps the most serious of these is an unwillingness to treat the current generation of KBSs out of the context of the promises of AI. The current generation of KBSs are not, in general, capable of learning or even modestly adaptive. These systems exhibit few nondeterministic properties. These KBSs may be complex but they are not unpredictable. But so long as there is this persistence in dwelling on the ultimate potential of AI systems instead of on the realities of the system being qualified, it is unlikely that an AFSR panel would allow flight testing.

A further difficulty arises from the contention that KBSs do not always produce the correct answer. If this is true then a KBS can only be used for tasks in which their performance can be monitored and overridden by a human. Most operation-critical systems are required to perform without human intervention or with only high-level supervision or control. However, a KBS that does not always produce the optimum answer is acceptable as long as it never produces a wrong answer. This latter point is in fact one of the main V&V issues: operation-critical systems must be shown to always produce acceptable solutions.

3.3 A Proposed Approach to Develop a Verification and Validation Methodology for Knowledge-Based Systems

In order to validate a system, one must have a set of requirements for that system, and those requirements must establish the performance criteria and the limitations of the system. The cur-

rent claim from some within the AI community that many of the characteristics of AI systems preclude such requirements either do not understand the validation issue or are unwilling to accept the structure and formalism required for validation. To address the issue of requirements, an incremental approach to validating KBSs is needed.

There are two key aspects of the proposed approach to the V&V of KBSs:

1. development of a KBS to perform some task that is well-known, well-understood, and for which conventional V&V techniques are adequate; and

2. incrementally and simultaneously expand both the KBS and the V&V techniques to more demanding and complex tasks.

The procedures used for verifying, qualifying, and validating conventional operation-critical flight systems at Ames-Dryden will be applied and modified as required. Because we ultimately plan to carry these experiments to flight using the rapid-prototyping facility [20], this process will be performed under the aegis of the AFSR panel and will be under periodic review. The subject of this research will be a KBS that is being developed to perform aircraft maneuvers normally performed by highly trained pilots.

The research plan is to identify maneuvers of increasing difficulty and to build gradually more complex and adaptive KBS to accomplish those maneuvers. This will include prototyping, evaluation, and a series of initial operating capabilities that will evolve into a sequence of documented requirements for testing against each version of the system. This approach fits well within the model of and practice used with conventional digital systems.

4 VALIDATING A SIMPLE KNOWLEDGE-BASED SYSTEM

To illustrate the proposed approach to the V&V of KBSs, a rule-based longitudinal altitude-command autopilot example for an F-15 aircraft will be presented. The example presented represents a single axis of a three-axis (longitudinal, lateral-directional, and velocity axes) controller. This controller is being developed and will be qualified as a mission-critical system (see section 2.3) as part of the research into validation methodologies for operation-critical KBSs.

4.1 Goals and Requirements for Example Knowledge-Based System

A simplified representation of the aircraft and control system is shown in figure 6. The objective is to develop and to demonstrate a knowledge-based controller that produces command inputs to the aircraft control system based on a dynamic world model obtained from instruments on the aircraft and on a simple set of rules. While this task

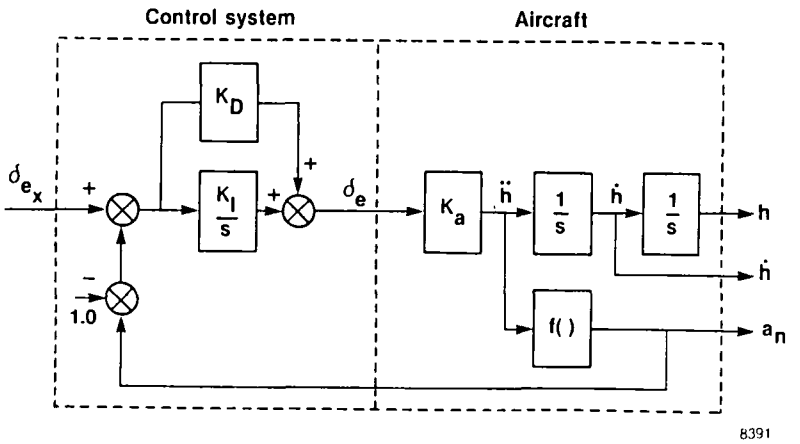


Figure 6. Simplified Longitudinal Model of the F-15 Aircraft and Its Control System

may not represent a suitable application of a KBS (because it is easily performed by conventional algorithmic control laws), it provides a simple mission-critical application that is both easy to understand and easy to validate.

The control task requires the autopilot system (whether based on conventional algorithms or a knowledge-based approach) to produce commands that cause the measured aircraft altitude h to be within some specified tolerance Δh of the commanded altitude h_{com}. Additionally, constraints are placed on the altitude rate $\dot{h}$ and the normal acceleration a_n. The constraint on a_n is the same as a constraint on altitude acceleration $\ddot{h}$, but a_n represents a more easily understood and easily measured physical quantity.

The initial requirement for this controller was that it control the aircraft in a consistent, repeatable manner at least as well as a pilot during both the transition mode (going from one altitude to another) and the altitude-hold mode (controlling the aircraft about a specified altitude). The desire was to have it control the aircraft as well as a conventional algorithmic autopilot. An additional goal was to allow off-condition engagement so that the controller would be effective even without benign initial (engagement) conditions.

These goals and requirements are similar to those initially imposed on the altitude-hold capabilities of the flight test maneuver autopilot for the IIiMAT vehicle [21]. The constraints and tolerances were established as baseline figures. From this initial specification, a rule-based system was implemented that combined numeric and symbolic methods. This initial system was tested using a detailed nonlinear simulation model of the aircraft and its control system; the controller achieved excellent results for some initial conditions but performed poorly for many others. This initial result was typical of that experienced when evaluating the initial implementation of a conventional controller on a nonlinear simulation. After several iterations of this process, a fairly detailed statement of performance capabilities and limitations was established (table I). This information, in essence, represents a clarification of the statement of goals and requirements, serves as the basis of a functional specification for the system, and defines the system test matrix.

4.2 Life Cycle of Example Knowledge-Based System

By this point in the life cycle, the development of a conventional controller would be supported by design and analysis tools and abstract (linear) models of not only the aircraft and its control system but of the controller as well. These tools and models would provide some of the basis of the validation of a conventional system by establishing metrics of system performance and robustness. The main benefit of having such tools and models is that their use allows extensive testing with a minimum of computational expense; only selected test points need to be repeated using the nonlinear simulation. For the rule-based controller, tools and analysis techniques either do not exist or are rudimentary at best. This difference in development will create some difficulties in qualifying the system for flight. Parts of the problem are both technical and sociological. Verification will have to rely on more extensive testing and a thorough exposition of the nature of the rules. The testing will require that a large number of tests be conducted on the nonlinear simulation that extends the time required for conducting those tests.

Table I. System Performance Capabilities and Engagement Conditions Defined by Prototyping

Performance requirements		
Δh	=	$\pm$ 50 ft
$\dot{h}_{max}$	=	$\pm$ 100 ft/sec
$a_{n_{pos}}$	=	2.0 g
$a_{n_{neg}}$	=	0.5 g
Engagement conditions		
Δh	=	$\pm\infty$ ft
$\dot{h}_{max}$	=	$\pm$ 200 ft/sec
a_n	=	$\pm$ 2.0 g

The next step in the life cycle is an SDR. This has been conducted informally during development but now requires formal exposure and review. The rules derived from prototyping (table II) and a detailed definition of the verification

test matrix will be presented and reviewed at the SDR. Again, this addresses both the technical and sociological aspects of V&V: the SDR provides a technical assessment of the design, allowing the completeness and consistency of the rules to be examined by independent reviewers and serves as a gentle introduction to the idea of using KBSs in such applications.

Table II. Rules for Longitudinal Altitude-Hold Autopilot

Performance boundary rules*
• If the altitude acceleration exceeds the positive acceleration limit, move stick forward.
• If the altitude acceleration exceeds the negative acceleration limit, move stick aft.
• If the predicted altitude rate exceeds the positive altitude rate limit, trim stick forward.
• If the predicted altitude rate exceeds the negative altitude rate limit, trim stick aft.
Normal command rules*
• If the altitude error is positive and the predicted altitude rate is negative, trim stick aft.
• If the altitude error is negative and the predicted altitude rate is positive, trim stick forward.
• If the predicted altitude error is positive and the altitude error is small, click stick forward.
• If the predicted altitude error is negative and the altitude error is small, click stick aft.
• If the predicted altitude error is positive and the altitude error is large, trim stick forward.
• If the predicted altitude error is negative and the altitude error is large, trim stick aft.

*Definitions:

move	large movement of stick
trim	intermediate movement of stick
click	small movement of stick

It is expected that the development of this rule-based controller will continue through the normal life cycle for research systems. The main differences that are expected between conventional and KBSs are that for the KBS

1. the design reviews will serve both educational and technical purposes;

2. the design will incorporate more problem specific experience (but probably less fundamental system understanding) at each stage in the life cycle;

3. the lack of traditional tools and abstract models will force earlier recognition and definition of system testing requirements; and

4. because of the lack of tools and abstract models, the testing required for the rule-based system will be more extensive than that required for a conventional system of similar capabilities.

4.3 Test Matrix for Example Knowledge-Based System

To appreciate the number of individual tests that must be performed as part of the validation of this longitudinal autopilot, two factors must be understood:

1. the performance and limitations define a matrix of test conditions for each simulated flight condition; and,

2. because the dynamics of an aircraft vary throughout its flight envelope, that matrix of test points must be repeated at many flight conditions.

The performance requirements and engagement conditions define the requirements for both on- and off-condition operation. To test the on-condition requirements for the example autopilot, one engages the system at the test altitude and Mach number and monitors the performance of the system to ensure that none of the performance limits are exceeded. The testing of engagement requirements requires a set of tests about each of the altitude and Mach number points. Thus, for a given altitude and Mach number, the system must be engaged at a number of conditions representing

the permutations of the bounds of the engagement conditions; again, time histories are monitored to ensure that the system performs within the limits established by the performance requirements. At each altitude and Mach number test condition, this requires a minimum of eight separate tests.

The dynamics of an aircraft are not constant throughout the flight envelope. To ensure that the system performance goals are met, tests must be performed at a number of flight conditions (figure 7). At each altitude and Mach number condition, the entire matrix of performance requirements must be tested at the engagement limits.

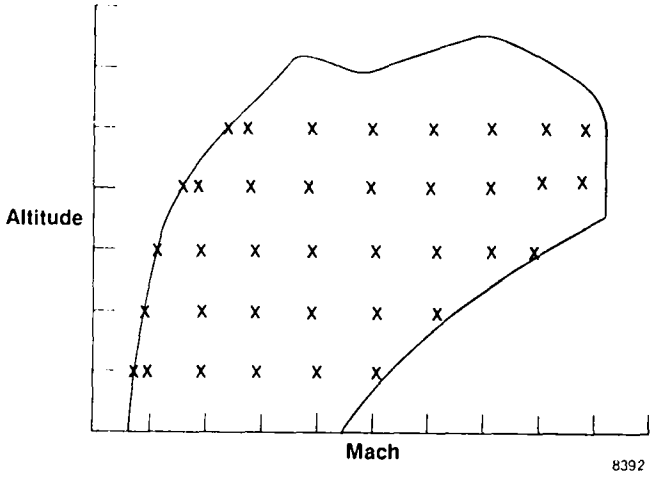


Figure 7. Typical Flight Envelope With Example Test Conditions

This testing is time consuming and requires a detailed nonlinear simulation. A conventional system would require less simulation testing on the nonlinear simulation because it would be supported by abstract models of the aircraft and the autopilot. The nonlinear simulation would be used at a few selected altitude and Mach number conditions to verify the abstract models.

It is important to note that the testing described above

1. includes no failure condition testing,

2. the example autopilot is a greatly simplified representation of a system that will be taken to flight, and

3. the rules presented in table II represent only a single axis of a three-axis controller.

5 LIMITATIONS OF VALIDATION METHODOLOGY FOR KNOWLEDGE-BASED SYSTEMS

The most serious limitation of applying the V&V methodology for conventional systems to operation-critical KBSs is the lack of both structured development methods and verification tools and techniques. Conventional systems are supported by design and analysis tools and techniques, coding standards, and methods for examining software that is procedural in nature. These tools, standards, and procedures do not exist for KBSs nor are any likely to emerge in the near term. Another limitation of applying the conventional V&V methodology to KBSs is that component testing is difficult if not impossible. Both of these limitations will force validation to rely on integrated system testing, treating the total KBS as a black box.

The testing requirements for a system do not increase linearly with the complexity of the system; testing requirements grow as a polynomial or exponential function of system complexity. As a simple example of the growth of the test matrix with system complexity, consider the test matrix defined for the longitudinal autopilot (see section 4.3). A similar matrix would be defined for each axis of the total autopilot. If we assume that there are m tests required for each axis, then the final autopilot will have three axes of comparable complexity; the total number of tests will be m^3 because all combinations of tests will have to be performed at each flight condition to validate the system performance.

Testing any but the most simple systems as black boxes requires a test matrix of overwhelming complexity. This will compound an already severe problem that has been a consistent factor in the V&V of conventional systems: the cost, schedule, and personnel requirements for V&V greatly exceed the development costs and almost always cause programmatic delays. Further, the costs and delays are directly related to how late, in the de-

velopment cycle, design and implementation errors are detected.

One of the main challenges of developing a validation methodology for KBSs is to develop tools and techniques that will allow highly complex systems to be verified, qualified for flight, and validated in a cost-effective and timely manner without having to reduce the capability or operational envelope of that system. (This challenge, incidentally, is one that those working with conventional systems must also face.) As part of the Ames-Dryden effort in developing and demonstrating a viable validation methodology for KBSs, the development of automatic testing systems is an integral part. The goal of this effort is to generate test matrices automatically from requirements and specifications for use in an automated testing system capable of both conducting tests and monitoring and interpreting test results.

6 CONCLUDING REMARKS

The qualification, verification, and validation methodology used at Ames-Dryden for flight-critical control systems and how this methodology can be extended and applied to intelligent knowledge-based systems are reviewed in this paper. The justification for the use of this methodology is the similarity of the current generation of KBSs with conventional systems in terms of complexity and function. Limitations of the proposed methodology for both highly reliable, fault-tolerant systems and extremely complex systems such as might be envisioned for future generations of KBSs are discussed. Research and development areas are suggested to augment and enhance the current methodology to support both conventional systems as well as KBSs.

The main differences between conventional systems and KBSs are that for the latter

1. the design reviews will serve both educational and technical purposes,

2. the design will incorporate more problem-specific experience (but probably less fundamental system understanding) at each stage in the life cycle,

3. the lack of traditional tools and abstract models will force earlier recognition and definition of system testing requirements, and

4. because of the lack of tools and abstract models, the testing required for the rule-based system will be more extensive than that required for a conventional system of similar capabilities.

The view presented in this paper is consistent with that proposed in Gault and others [1]:

> A validation methodology for such systems [ultrahigh reliability, fault-tolerant systems] must be based on *a judicious combination of logical proofs, analytical modeling, and experimental testing.*

This methodology must be supported by reliable, validated development and test tools that lower the cost and reduce the schedule, if the goal of validation is to be achieved for either highly reliable, fault-tolerant systems or highly complex systems such as are envisioned for KBSs.

Perhaps the biggest obstacle in the qualification of operation-critical KBSs is the mystification and obfuscation by the advocates and developers of KBSs. While stressing the enormous differences between KBSs and conventional systems may be a useful tactic in generating enthusiasm and support for the development and use of KBSs, this approach is almost guaranteed to discourage acceptance and prevent deployment of these systems in operation-critical applications.

REFERENCES

1. Gault, J.W., Trivedi, K.S., and Clary, J.B, Eds., "Validation Methods Research for Fault-Tolerant Avionics and Control Systems — Working Group Meeting II," NASA CP-2130, 1980.

2. "Description and Flight Test Results of the NASA F-8 Digital Fly-By-Wire Control System — A Collection of papers from the NASA Symposium on Advanced Control Technology, Los Angeles, California, July 9-11, 1974," NASA TN D-7843, 1975.

3. Szalai, K.J., Felleman, P.G., Gera, Joseph, and Glover, R.D., "Design and Test Experience with a Triply Redundant Digital Fly-By-Wire Control System," AIAA-77-1042, 1977.

4. Szalai, K.J., Jarvis, C.R., Krier, G.E., Megna, V.A., Brock, L.D., and O'Donnell, R.N., "Digital Fly-By-Wire Flight Control Validation Experience," NASA TM-72860, 1978.

5. Edwards, J.W., and Deets, D.A., "Development of a Remote Digital Augmentation System and Application to a Remotely Piloted Research Vehicle," NASA TN D-7941, 1975.

6. Evans, M.B., and Schilling, L.J., "The Role of Simulation in the Development and Flight Test of the HiMAT Vehicle," NASA TM-84912, 1984.

7. Myers, A.F., and Sheets, S.G., "Qualification of HiMAT Flight Systems," 7th Annual Technical Symposium, Association for Unmanned Vehicle Systems Proceedings, Dayton, Ohio, June 1980.

8. Myers, A.F., Earls, M.R., and Callizo, L.A., "HiMAT Onboard Flight Computer System Architecture and Qualification," Journal of Guidance, Control, and Dynamics, Vol. 6, No. 4, 1983, pp. 231–238.

9. Petersen, K.L., "Flight Control Systems Development of Highly Maneuverable Aircraft Technology (HiMAT) Vehicle," AIAA-79-1789, Aug. 1979.

10. Mackall, D.A., Ishmael, S.D., and Regenie, V.A., "Qualification of the Flight-Critical AFTI/F-16 Digital Flight Control System," AIAA-83-0060, Jan. 1983.

11. Mackall, D.A., Regenie, V.A., and Gordoa, Michael, "Qualification of the AFTI/F-16 Digital Flight Control System," NAECON Paper 324, May 1983.

12. Holt, H.M., Lupton, A.O., and Holden, D.G., "Flight Critical System Design Guidelines and Validation Methods," AIAA-84-2461, Nov. 1984.

13. Hopkins, Jr., A.L., "General Validation Issues," *in* "Validation Methods for Fault-Tolerant Avionics and Control Systems — Working Group Meeting I," NASA CP-2114, 1979, pp. 27–30.

14. Hartmann, G.L., Wall, Jr., J.E., and Rang, E.R., "Design Validation of Fly-By-Wire Flight Control Systems," *in* "Fault-Tolerant Hardware/Software Architecture for Flight Critical Functions," AGARD Lecture Series No. 143, 1985.

15. Gerhart, S.L., "Limitations of Proving and Testing," *in* "Validation Methods for Fault-Tolerant Avionics and Control Systems — Working Group Meeting I," NASA CP-2114, 1979, pp. 47–53 and 85–88.

16. Petersen, K.L., and Flores, Jr., Christobal, "Software Control and System Configuration Management: A Systems-Wide Approach," NASA TM-85908, 1984.

17. Culbert, Chris, Riley, Gary, and Savely, R.T., "Approaches to the Verification of Rule-Based Expert Systems," SOAR '87, First Annual Workshop on Space Operations Automation and Robotics, Aug. 1987.

18. Richardson, Keith, and Wong, Carla, "Knowledge Based System Verification and Validation as Related to Automation of Space Station Subsystems: Rationale for a Knowledge Based System Lifecycle," *in* "Conference Proceedings of the Second Annual Artificial Intelligence Research Forum," Information Sciences Division, NASA Ames Research Center, Moffett Field, California, Nov. 1987, pp. 153–158.

19. Stachowitz, R.A., Combs, J.B., and Chang, C.L., "Validation of Knowledge-Based Systems," AIAA-87-1685, Mar. 1987.

20. Duke, E.L., Regenie, V.A., and Deets, D.A., "Rapid Prototyping Facility for Flight Research in Artificial-Intelligence-Based Flight Systems Concepts," NASA TM-88268, 1986.

21. Duke, E.L., Jones, F.P., and Roncoli, R.B., "Development and Flight Test of an Experimental Maneuver Autopilot for a Highly Maneuverable Aircraft," NASA TP-2618, 1986.

NASA
National Aeronautics and Space Administration

Report Documentation Page

1. Report No.	2. Government Accession No.	3. Recipient's Catalog No.
NASA TM-100442		

4. Title and Subtitle

Application of Flight Systems Methodologies to the Validation of Knowledge-Based Systems

5. Report Date

July 1988

6. Performing Organization Code

7. Author(s)

Eugene L. Duke

8. Performing Organization Report No.

H-1493

10. Work Unit No.

RTOP 505-66-11

9. Performing Organization Name and Address

NASA Ames Research Center
Dryden Flight Research Facility
P.O. Box 273, Edwards, CA 93523-5000

11. Contract or Grant No.

13. Type of Report and Period Covered

Technical Memorandum

12. Sponsoring Agency Name and Address

National Aeronautics and Space Administration
Washington, DC 20546

14. Sponsoring Agency Code

15. Supplementary Notes

Prepared for presentation at the Space Operations Automation and Robotics (SOAR) Workshop, July 20–23, 1988, at Dayton, Ohio.

16. Abstract

Flight and mission-critical systems are verified, qualified for flight, and validated using well-known and well-established techniques. These techniques define the validation methodology used for such systems. In order to verify, qualify, and validate knowledge-based systems (KBSs), the methodology used for conventional systems must be addressed, and the applicability and limitations of that methodology to KBSs must be identified. The author presents an outline of how this approach to the validation of KBSs is being developed and used at the Dryden Flight Research Facility of the NASA Ames Research Center.

17. Key Words (Suggested by Author(s))

Artificial intelligence
Knowledge-based systems
Qualification
Verification and validation

18. Distribution Statement

Unclassified — Unlimited

Subject category 63

19. Security Classif. (of this report)	20. Security Classif. (of this page)	21. No. of pages	22. Price
Unclassified	Unclassified	16	A02

NASA FORM 1626 OCT 86

For sale by the National Technical Information Service, Springfield, VA 22161-2171.